

## DATA PROCESSOR

Publication number: JP8110804

Publication date: 1996-04-30

Inventor: ISHIKAWA HITOSHI

Applicant: OMRON TATEISI ELECTRONICS CO

Classification:

- international: G06F15/16; G05B19/05; G06F9/50; G06F15/177;  
G06F15/16; G05B19/05; G06F9/46; (IPC1-7):  
G05B19/05; G06F15/16

- European:

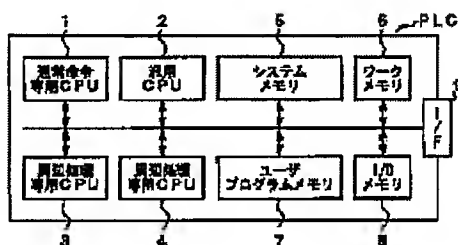
Application number: JP19940245118 19941011

Priority number(s): JP19940245118 19941011

Report a data error here

### Abstract of JP8110804

**PURPOSE:** To effectively utilize CPU resources by providing a CPU exclusive for normal instructions and a general CPU.  
**CONSTITUTION:** When the CPU 1 exclusive for normal instructions reads out a special instruction while reading out a user program, a request flag indicating that the special instruction is read out is set in the flag area, etc., of a work memory 6. A CPU 2 checks the request flag throughout program execution to judge whether or not the request flag is set, and executes the requested special instruction when the request flag is set. After the special instruction is executed, the general CPU 2 sends a report on the completion of processing to the CPU 1 exclusive for normal instruction to make the CPU 1 exclusive for normal operation execute the normal instructions, and it is judged whether or not the program execution is completed by one scan. When the execution has been completed by one scan, peripheral processing is performed successively.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-110804

(43)公開日 平成8年(1996)4月30日

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 5 B 19/05				
G 0 6 F 15/16	3 7 0 Z		G 0 5 B 19/ 05	F

審査請求 未請求 請求項の数5 O L (全 11 頁)

(21)出願番号 特願平6-245118

(22)出願日 平成6年(1994)10月11日

(71)出願人 000002945

オムロン株式会社

京都府京都市右京区花園土堂町10番地

(72)発明者 石川 仁

京都府京都市右京区花園土堂町10番地 オムロン株式会社内

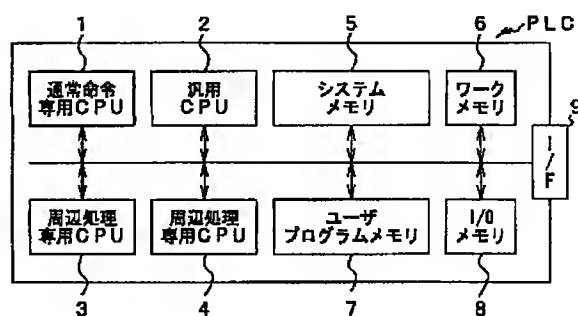
(74)代理人 弁理士 和田 成則

(54)【発明の名称】 データ処理装置

(57)【要約】

【目的】 CPUパワーを有効に利用すると共に、サイクルタイムを短縮化する。

【構成】 通常命令専用CPU 1は、ユーザプログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合にはフラグのセットにより当該特別命令の実行依頼を出力する。汎用CPU 2は、通常命令専用CPUによるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行し、上記プログラムの1スキャン終了後に、周辺処理を実行する。



## 【特許請求の範囲】

【請求項 1】 通常命令および特別命令からなるプログラムを複数の CPU により実行すると共に、周辺処理を実行するデータ処理装置において、

上記プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には当該特別命令の実行依頼を出力する通常命令専用 CPU と、

上記周辺処理を実行すると共に、上記通常命令専用 CPU から特別命令の実行依頼が出力された場合には当該特別命令を実行する汎用 CPU と、

を具備することを特徴とするデータ処理装置。

【請求項 2】 通常命令専用 CPU は、プログラム中から各命令を順次読出し、特別命令を読出した場合にはフラグのセットにより当該特別命令の実行依頼を出力し、

汎用 CPU は、上記通常命令専用 CPU によるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行して、上記プログラムの 1 スキャン終了後に周辺処理を実行する、

ことを特徴とする請求項 1 記載のデータ処理装置。

【請求項 3】 通常命令専用 CPU は、プログラム中から各命令を順次読出し、特別命令を読出した場合には割込み信号の出力により当該特別命令の実行依頼を出力し、

汎用 CPU は、通常は周辺処理を実行し、上記通常命令専用 CPU から割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する、

ことを特徴とする請求項 1 記載のデータ処理装置。

【請求項 4】 通常命令および特別命令からなるプログラムを通常命令専用 CPU および汎用 CPU により実行すると共に、周辺処理を汎用 CPU および周辺処理専用 CPU により実行するデータ処理装置において、

上記汎用 CPU あるいは周辺処理専用 CPU に、上記プログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力するかを所定の評価項目に基づき推論して決定する推論手段を設け、

上記通常命令専用 CPU は、

上記プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には上記推論手段の決定に基づきフラグのセットか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力し、

上記汎用 CPU は、

上記通常命令専用 CPU からの実行依頼の出力方式に

じ、上記通常命令専用 CPU によるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行して上記プログラムの 1 スキャン終了後に周辺処理を実行するか、あるいは通常は周辺処理を実行して上記通常命令専用 CPU から割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する、ことを特徴とするデータ処理装置。

【請求項 5】 通常命令および特別命令からなるプログラムを複数の CPU により実行すると共に、周辺処理を実行するデータ処理装置において、

上記プログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力するかをパラメータとして遺伝子を複数作成し、各遺伝子のパラメータに基づき上記プログラムおよび周辺処理をシミュレーションすると共に、各遺伝子の複製、交差、突然変異を繰返して、評価の高い遺伝子を決定するシミュレーション手段と、

上記プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には上記シミュレーション手段によって決定された評価の高い遺伝子のパラメータに基づきフラグのセットか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力する通常命令専用 CPU と、上記通常命令専用 CPU からの実行依頼の出力方式に依り、上記通常命令専用 CPU によるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行して上記プログラムの 1 スキャン終了後に周辺処理を実行するか、あるいは通常は周辺処理を実行して上記通常命令専用 CPU から割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する汎用 CPU と、を具備することを特徴とするデータ処理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、プログラムを複数の CPU により実行すると共に、周辺処理を実行するようにしたデータ処理装置に関する。

## 【0002】

【従来の技術】近年、高速処理を目的として、例えば図 10 に示すように、複数のプログラム専用 CPU A、B および周辺処理専用 CPU C、D により周辺処理およびプログラム実行処理を各々並列処理するデータ処理装置が開発されている。

## 【0003】

【発明が解決しようとする課題】しかし、このようなデータ処理装置では、周辺処理は並列処理が行い易いため各周辺処理専用 CPU C、D は並列に動作可能であるが、プログラム実行処理は通常並列処理が行えず、プロ

グラム実行専用CPU Aが動作しているときには、プログラム実行専用CPU Bが停止しているため、CPU資源が無駄になる、という問題があった。

【0004】また、周辺処理の実行中は通常、プログラム専用CPU A、Bが停止しているため、この点でもCPU資源が無駄になる、という問題があった。

【0005】そこで、本発明は、CPU資源を有効に利用できるデータ処理装置を提供することを目的とする。

【0006】

【課題を解決するための手段】上記目的を達成するため、請求項1記載の発明では、通常命令および特別命令からなるプログラムを複数のCPUにより実行すると共に、周辺処理を実行するデータ処理装置において、上記プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には当該特別命令の実行依頼を出力する通常命令専用CPUと、上記周辺処理を実行すると共に、上記通常命令専用CPUから特別命令の実行依頼が出力された場合には当該特別命令を実行する汎用CPUと、を具備することを特徴とする。

【0007】請求項2記載の発明では、請求項1記載のデータ処理装置において、通常命令専用CPUは、プログラム中から各命令を順次読出し、特別命令を読出した場合にはフラグのセットにより当該特別命令の実行依頼を出力し、汎用CPUは、上記通常命令専用CPUによるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行して、上記プログラムの1スキャン終了後に周辺処理を実行する、ことを特徴とする。

【0008】請求項3記載の発明では、請求項1記載のデータ処理装置において、通常命令専用CPUは、プログラム中から各命令を順次読出し、特別命令を読出した場合には割込み信号の出力により当該特別命令の実行依頼を出力し、汎用CPUは、通常は周辺処理を実行し、上記通常命令専用CPUから割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する、ことを特徴とする。

【0009】請求項4記載の発明では、通常命令および特別命令からなるプログラムを通常命令専用CPUおよび汎用CPUにより実行すると共に、周辺処理を汎用CPUおよび周辺処理専用CPUにより実行するデータ処理装置において、上記汎用CPUあるいは周辺処理専用CPUに、上記プログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力するかを所定の評価項目に基づき推論して決定する推論手段を設け、上記通常命令専用CPUは、上記プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には上記推論手段の決定に

基づきフラグのセットか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力し、上記汎用CPUは、上記通常命令専用CPUからの実行依頼の出力方式に応じ、上記通常命令専用CPUによるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行して上記プログラムの1スキャン終了後に周辺処理を実行するか、あるいは通常は周辺処理を実行して上記通常命令専用CPUから割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する、ことを特徴とする。

【0010】請求項5記載の発明では、通常命令および特別命令からなるプログラムを複数のCPUにより実行すると共に、周辺処理を実行するデータ処理装置において、上記プログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力するかをパラメータとして遺伝子を複数作成し、各遺伝子のパラメータに基づき上記プログラムおよび周辺処理をシミュレーションすると共に、各遺伝子の複製、交差、突然変異を繰返して、評価の高い遺伝子を決定するシミュレーション手段と、上記プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には上記シミュレーション手段によって決定された評価の高い遺伝子のパラメータに基づきフラグのセットか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力する通常命令専用CPUと、上記通常命令専用CPUからの実行依頼の出力方式に応じ、上記通常命令専用CPUによるプログラムの実行中は上記フラグをチェックし上記フラグがセットされていれば特別命令を実行して上記プログラムの1スキャン終了後に周辺処理を実行するか、あるいは通常は周辺処理を実行して上記通常命令専用CPUから割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する汎用CPUと、を具備することを特徴とする。

【0011】

【作用】請求項1記載の発明では、通常命令専用CPUは、プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には当該特別命令の実行依頼を出力し、汎用CPUは、周辺処理を実行すると共に、通常命令専用CPUから特別命令の実行依頼が出力された場合には当該特別命令を実行する。

【0012】請求項2記載の発明では、請求項1記載のデータ処理装置において、通常命令専用CPUは、プログラム中から各命令を順次読出し、特別命令を読出した場合には、フラグのセットにより当該特別命令の実行依頼を出力し、汎用CPUは、通常命令専用CPUによる

プログラムの実行中は上記フラグをチェックして、上記フラグがセットされていれば特別命令を実行し、上記プログラムの1スキャン終了後に周辺処理を実行する。

【0013】請求項3記載の発明では、請求項1記載のデータ処理装置において、通常命令専用CPUは、プログラム中から各命令を順次読出し、特別命令を読出した場合には、割込み信号の出力により当該特別命令の実行依頼を出力し、汎用CPUは、通常は周辺処理を実行して、通常命令専用CPUから割込信号の出力により特別命令の実行依頼が出力された場合には、その割込信号を受けて当該特別命令を実行する。

【0014】請求項4記載の発明では、まず、通常命令専用CPUあるいは汎用CPUに設けられた推論手段がプログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力するかを所定の評価項目に基づき推論して決定する。

【0015】そして、通常命令専用CPUは、プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合には推論手段の決定に基づきフラグのセットか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力する。汎用CPUは、通常命令専用CPUからの実行依頼の出力方式に応じ、通常命令専用CPUによるプログラムの実行中は上記フラグをチェックして、上記フラグがセットされていれば特別命令を実行し、プログラムの1スキャン終了後に周辺処理を実行するか、あるいは通常は周辺処理を実行して、上記通常命令専用CPUから割込信号の出力により特別命令の実行依頼が出力された場合にはその割込信号を受けて当該特別命令を実行する。

【0016】請求項5記載の発明では、まず、シミュレーション手段は、プログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力するかをパラメータとして遺伝子を複数作成し、各遺伝子のパラメータに基づき上記プログラムおよび周辺処理をシミュレーションすると共に、各遺伝子の複製、交差、突然変異を繰返して、評価の高い遺伝子を決定する。

【0017】そして、通常命令専用CPUは、プログラム中から各命令を順次読出し、通常命令を読出した場合には当該通常命令を実行する一方、特別命令を読出した場合にはシミュレーション手段によって決定された評価の高い遺伝子のパラメータに基づきフラグのセットか、あるいは割込み信号の出力により当該特別命令の実行依頼を出力する。汎用CPUは、通常命令専用CPUからの実行依頼の出力方式に応じ、通常命令専用CPUによるプログラムの実行中は上記フラグをチェックして、上

記フラグがセットされていれば特別命令を実行し、プログラムの1スキャン終了後に周辺処理を実行するか、あるいは通常は周辺処理を実行して、通常命令専用CPUから割込信号の出力により特別命令の実行依頼が出力された場合には、その割込信号を受けて当該特別命令を実行する。

【0018】

【実施例】以下、本発明に係るデータ処理装置の実施例として、プログラマブルコントローラ（以下「PLC」という。）の各実施例を図面に基いて説明する。

【0019】図1は、本発明に係るPLCの構成を示している。

【0020】このPLCは、ユーザプログラムを読出してユーザプログラム中のほとんどの命令である通常命令、例えば基本命令等を実行等する通常命令専用CPU1と、後述するようにユーザプログラム中の特別命令、例えば応用命令等を実行すると共に周辺処理も実行する汎用CPU2と、周辺処理のみを実行する周辺処理専用CPU3、4と、システムメモリ5と、ワークメモリ6と、ユーザプログラムメモリ7と、I/Oメモリ8と、I/O機器（図示せず。）等の外部接続機器が接続されるI/F9とを有している。

【0021】プログラム実行専用CPU1および汎用CPU2は、第1実施例では、後述するポーリング方式により特別命令の実行依頼および実行処理を行うように構成されている。

【0022】また、汎用CPU2および周辺処理専用CPU3、4は、各々、各周辺処理の内容に応じて予め定められた、あるいは必要に応じて選択された各周辺処理を実行するように構成されている。

【0023】次に、本発明に係るPLCの各実施例の動作を図面を参照して説明する。

【0024】図2は、第1実施例のポーリング方式による汎用CPU2側の処理を示している。

【0025】まず、この図では示していないが、通常命令専用CPU1が、ユーザプログラムの読出中に特別命令を読出した場合には、ワークメモリ6のフラグエリア等に特別命令を読出したことを示す依頼フラグをセットする。

【0026】汎用CPU2は、プログラム実行中は、常時、その依頼フラグをチェックして（ステップ100）、依頼フラグがセットされているか否かを判断し（ステップ110）、依頼フラグがセットされている場合には（ステップ110 “Yes”）、その依頼された特別命令の実行処理を行う（ステップ120）。

【0027】そして、その特別命令の実行終了後、汎用CPU2は、通常命令専用CPU1に処理完了通知を送り（ステップ130）、通常命令専用CPU1に通常命令を実行させ、プログラム実行が1スキャン分終了する否かを判断する（ステップ140）。

【0028】ここで、プログラムの実行が1スキャン分終了していない場合は(ステップ140“N o”)、上記ステップ100~140までの処理を繰り返す。一方、プログラムの実行が1スキャン分終了した場合は(ステップ140“Y es”)、続いて周辺処理を行い(ステップ150)、この周辺処理が終了すればまた上記ステップ100からの処理を繰り返す。

【0029】従って、この第1実施例によれば、汎用CPU2はプログラム実行中は依頼フラグを常時チェックしに行き、依頼フラグがセットされていれば特別命令を実行する一方、プログラム実行が終了すれば周辺処理も実行するようにしたため、CPU資源を有効に利用できることになる。

【0030】次に、本発明の第2実施例を説明する。

【0031】第2実施例に係るPLCでは、通常命令専用CPUがプログラム中から特別命令を讀出して、当該特別命令の実行依頼の出力を、第1実施例のポーリング方式とは異なり、後述するインタラプト(割込み)方式により実行依頼を出力するように構成したものである。

【0032】なお、この第2実施例に係るPLCの構成では、図1に示す第1実施例に係るPLCの構成の同様であるため、その説明は省略して、図1に基づいて第2実施例のインタラプト方式による実行依頼の出力動作について説明する。

【0033】図3は、第2実施例のインタラプト方式による汎用CPU2側の実行処理を示している。

【0034】この第2実施例の場合、通常、汎用CPU2は、通常命令専用CPU1から割込みがなければ(ステップ200“N o”)、プログラムの実行中にかかわらず、処理すべき周辺処理があればその周辺処理を実行する(ステップ210)。

【0035】そして、この図では示していないが、通常命令専用CPU1が、ユーザプログラムの読出し中に特別命令を讀出した場合には、その特別命令の実行依頼として割込み信号を汎用CPU2に向けて送る。

【0036】汎用CPU2は、通常命令専用CPU1から\*

1回のインタラプト発生に伴うオーバーヘッド	$T_o$ (ns)
通常命令専用CPU1で実行する1命令の平均実行時間	$T_a$ (ns)
汎用CPU2で実行する1命令の平均実行時間	$T_b$ (ns)
通常命令専用CPU1で1サイクルに実行する平均命令数	$N_a$ (個)
汎用CPU2で1サイクルに実行する平均命令数	$N_b$ (個)
1サイクルに処理される周辺処理の平均的負荷	$T_p$ (ns)

このとき、ポーリング方式のサイクルタイム $T_{pol}$ 、インタラプト方式のサイクルタイム $T_{int}$ は、それぞれ、下記のように計算される。ただし、周辺処理は全てプログラム実行中でも並列に処理できるものとする。

【0044】

$$T_{pol} = N_a T_a + N_b T_b + T_p$$

$$T_{int} = N_a T_a + N_b (T_b + T_o) + (T_p - N_a T_a)$$

\*ら割込みが送られてきた場合(ステップ200“Y es”)、通常命令専用CPU1が讀出した特別命令を実行して(ステップ220)、その特別命令実行終了後、通常命令専用CPU1へ完了通知を送り(ステップ230)、その後、周辺処理を行う(ステップ210)。

【0037】従って、この第2実施例によれば、第1実施例の場合と同様に、汎用CPU2がユーザプログラム中の特別命令の実行の他に、周辺処理の実行も行うため、CPU資源を有効に利用できる。

【0038】また、この第2実施例では、通常命令専用CPU1と汎用CPU2間の実行依頼の通信方式に割込み信号によるインタラプト方式を採用しているため、ユーザプログラムの実行中であっても、汎用CPU2は特別命令を実行してなければ周辺処理を実行することができるため、よりCPU資源を有効に利用できる。

【0039】次に、上記第1実施例のポーリング方式と、第2実施例のインタラプト方式とを比較して、その長所及び短所を説明する。

【0040】第1実施例のポーリング方式を採用した場合には、プログラム実行中にインタラプトオーバーヘッドが生じないという長所を有する一方、プログラム実行中に汎用CPU2が周辺処理を実行できないという短所を有するのに対し、第2実施例のインタラプト方式を採用した場合には、その反対で、プログラム実行中に汎用CPU2が周辺処理を実行できるという長所を有するのに対し、プログラム実行中にインタラプト(割込み)オーバーヘッドが生じるという短所を有することになる。

【0041】具体的に、CPU間の実行依頼の通信方式として、ポーリング方式と、インタラプト方式とを用いた場合のサイクルタイムの概算を示すと、下記の通りとなる。尚、簡略化のため、ポーリング方式の場合のフラグのチェックタイムや、完了通知による時間等は無視して説明する。

【0042】まず、下記のように各種時間を定義する。

【0043】

$$= N_b T_b + T_p + N_b T_o$$

従って、上記2式より、“ $N_a T_a$ ”と“ $N_b T_o$ ”の大小関係により、ポーリング方式と、インタラプト方式のいずれが優位かがわかる。

【0045】つまり、1回のインタラプト発生に伴うオーバーヘッド $T_o$ が大きいときや、汎用CPU2で1サイクルに実行する平均命令数 $N_b$ が多いときは、ポーリング方式のサイクルタイムが小となる。これに対し、通常

命令専用CPU1で1サイクルに実行する平均命令数N<sub>a</sub>が多いときや、通常命令専用CPU1で実行する1命令の平均実行時間T<sub>a</sub>が大きいとき、すなわち通常命令専用CPU1における負荷が大きいときは、インタラプト方式のサイクルタイムのほうが小さくなる。

【0046】よって、どちらの通信方式を選択するかは、複数の要因が複雑に関係し合うため、簡単に決定できない。

【0047】そこで、その選択方法としては、

(1) ユーザプログラム実行時に所定の評価項目を調べて、これをもとに推論してサイクルタイムの小さい通信方式に変更する方法。

【0048】(2) 実行前あるいは実行時に、シミュレーションを行うことにより、サイクルタイムの小さな通信方式を試行錯誤に構築していく方法。

【0049】の2つが考えられ、上記(1)の方法を採用したのが次に説明する第3実施例であり、上記(2)の方法を採用したのがその次に説明する第4実施例である。

【0050】次に、本発明の第3実施例を説明する。

【0051】この第3実施例は、上述の通り、実行時に所定の評価項目を調べ、これをもとに通常命令CPUおよび汎用CPU間の通信方式である第1実施例のポーリング方式および第2実施例のインタラプト方式を変更するように構成されており、通信方式の変更するための推論方法としてファジィ推論を採用し、ファジィ推論部を汎用CPUに設けていることを特徴としている。

【0052】図4は、この第3実施例に係るPLCの構成を示している。

【0053】なお、図4において汎用CPU2の具体的構成以外は、図1に示す第1実施例のPLCと同様であるため、同一符号を付して説明を省略し、汎用CPU2の具体的構成、およびその動作を説明する。

【0054】汎用CPU2は、ユーザプログラム中の特別命令を実行する特別命令実行部21と、周辺処理を実行する周辺処理部22と、後述するように所定の評価項目に基づき推論して、上記プログラム中から特別命令が読出された場合にはフラグのセットにより当該特別命令の実行依頼を出力するか、あるいは割込信号の出力により当該特別命令の実行依頼を出力するかを決定するファジィ推論部23と、を有している。

【0055】ファジィ推論部23におけるファジィ推論の際の所定の評価項目をルールで示すと、例えば、ルール1；プログラム実行中に、汎用CPU2が並列に行える処理が多い場合には、インタラプト方式が有利である。

【0056】ルール2；ユーザの要求として、周辺処理の性能を重視する場合には、インタラプト方式が有利である。

【0057】ルール3；ユーザプログラム中に、汎用C

PU2を利用するような命令が頻繁に発生する場合には、ポーリング方式のほうが有利である。

【0058】となる。

【0059】また、これをif-then式で書き替えると、

ルール1；if（汎用CPU2が並列に行える処理が多い）

then（評価＝インタラプト方式有利）

ルール2；if（要求される周辺処理の性能＝高速）

then（評価＝インタラプト方式有利）

ルール1；if（依頼先CPUの利用回数＝稀）

then（評価＝インタラプト方式有利）

となる。

【0060】図5(a)～(c)は、上記各ルール1～3のメンバーシップ関数を示している。

【0061】(a)は、ルール1のメンバーシップ関数を示しており、横軸には並列に行える処理数(個)を、縦軸にはインタラプト方式に対するマッチング度(0.0～1.0間の値)をとっている。

【0062】(b)は、ルール2のメンバーシップ関数を示しており、要求される周辺処理の性能を示しており、横軸には応答速度(秒)を、縦軸にはインタラプト方式に対するマッチング度をとっている。

【0063】(c)は、汎用CPUの利用回数のメンバーシップ関数を示しており、横軸には利用回数(回/秒)を、縦軸にはインタラプト方式に対するマッチング度をとっている。

【0064】従って、ファジィ推論部23では、評価項目である各ルール1～3について、各々のメンバーシップ関数の値から、直接インタラプト方式の優位性(0.0～1.0間の値)が求まるため、この平均を取り、この平均値が0.5より大きいならばインタラプト方式に、それより小さいならばポーリング方式にというように決定できる。

【0065】次に、この第3実施例の動作を図面を参照して説明する。

【0066】図6は、この第3実施例のファジィ推論によるCPU間実行依頼の決定方法を示している。

【0067】ファジィ推論部23は、まず通常命令専用CPU1や汎用CPU2の特別命令実行部21がユーザプログラムを実行したり、汎用CPU2の周辺処理部22や周辺処理専用CPU3、4が周辺処理を実行した結果から上記各評価項目の評価値を求める(ステップ300)。尚、初期状態においては、通常命令専用CPU1と汎用CPU2間の通信方式は、ポーリング方式か、あるいはインタラプト方式のどちらかに予め設定されているものとする。

【0068】そして、ファジィ推論部23は、その評価値を各々が対応する上記ルール1～3の各メンバーシップ関数に代入して、ファジィ推論を行い、その推論結果



すなわち各ルール1～3のマッチング度からその平均を求めて、インタラプト方式か、あるいはポーリング方式かの通信方式を決定して、決定した通信方式の指示を通常命令専用CPU1へ送る(ステップ310)。

【0069】通常命令専用CPU1は、その通信方式の指示を受けて、その通信方式に基づき汎用CPU2への実行依頼の通信方式を変更する(ステップ320)。

【0070】その後、通常命令専用CPU1は、ユーザプログラムから順次命令を読み出して、通常命令を読み出した場合は、通常命令を実行する一方、特別命令を読み出した場合は、ファジィ推論部23によって新たに決定された通信方式によりこの通常命令専用CPU1から汎用CPU2へ実行依頼を送出するというように、実行しているユーザプログラムに最適で、サイクルタイムの小さくなる通信方式により実行依頼を出力等して通常の処理を行う(ステップ330)。

【0071】従って、この第3実施例によれば、通常命令専用CPUと汎用CPU間の実行依頼の通信方式をポーリング方式で行うか、あるいはインタラプト方式で行うかをユーザプログラム実行時に所定の評価項目にファジィ推論によって推論してサイクルタイムの小さくなる通信方式を決定するようにしたため、ユーザプログラムに応じてユーザニーズにあった、サイクルタイムの小さくなる実行依頼の通信方式を動的に選択できると共に、CPU資源を有効に利用できる。

【0072】なお、この第3実施例では、汎用CPU2にファジィ推論部23を設けて説明したが、ファジィ推論部23を周辺処理専用CPU4あるいは通常命令専用CPU1に設けるようにしても良い。ただし、ファジィ推論部23を通常命令専用CPU1に設けると、通常命令専用CPU1による通常命令の実行速度が低下するため、なるべく汎用CPU2あるいは周辺処理専用CPU3、4にファジィ推論部23を設けるようにする。

【0073】また、この第3実施例では、汎用CPU2が1台の場合で説明したが、汎用CPU2が複数台あり、汎用CPU2が実行する特別命令の実行をどの汎用CPU2が行っても良いとする場合には、通常命令専用CPU1と汎用CPU2間の通信方式の決定の他に、どの汎用CPU2を選択するかをも決定する必要がある。

【0074】例えば、特別命令を読み出した場合にどの汎用CPU2を選択するかをファジィ推論によって決定するために、下記に示すようなルール4～6が考えられ、ルール4；現在、処理負荷が最も軽い汎用CPUがあれば、その汎用CPUを使用する。

【0075】ルール5；該当処理を高速に実行できる汎用CPUがあれば、その汎用CPUを利用する。

【0076】ルール6；他に実行できる処理の多い汎用CPUがあれば、その汎用CPUは利用しない。

【0077】このルール4～6をメンバーシップ関数で表わして、各ルールに基づいてファジィ推論を行えば、

多数ある汎用CPU2の中からその時点において最適な汎用CPU2を選択できることになる。

【0078】次に、本発明の第4実施例を説明する。

【0079】この第4実施例は、上述の通り、ユーザプログラムの実行前あるいは実行時にシミュレーションを導入することにより、サイクルタイムの小さくなるようなシステムを試行錯誤的に構築していき、これをもとにCPU間での実行依頼の通信方式である上記ポーリング方式およびインタラプト方式を変更する方法で、実行依頼の通信方式の変更に遺伝的アルゴリズム(GA)を採用している。

【0080】図7に、第4実施例に係るPLCの構成を示している。

【0081】この第4実施例に係るPLCは、各々が別の通常命令を実行する2台の通常命令専用CPU1a、1bと、各々別の特別命令を実行すると共に周辺処理も実行する2台の汎用CPU2a、2bとを有すると共に、各CPU間の実行依頼の通信方式を後述する遺伝的アルゴリズムを採用してシミュレーションすることにより、最もサイクルタイムの小さくなるCPU間実行依頼の通信方式を決定するシミュレーション装置10がインターフェース9を介して接続されていることを特徴とする。

【0082】尚、図1に示す第1実施例のPLCの構成要素と共通な構成要素には同一符号を付して説明を省略すると共に、周辺処理専用CPUは便宜上省略しているが、勿論設けられていても良い。

【0083】図8(a)、(b)は、各々、各CPU間の実行依頼の通信方式に遺伝的アルゴリズムを適用するためのモデルと、このモデルにおける遺伝子を示している。

【0084】(a)は、各CPU間の通信方式を示しており、通常命令専用CPU1aから通常命令専用CPU1bへの実行依頼の通信方式をWab、通常命令専用CPU1aから汎用CPU2aへの実行依頼の通信方式をWac、通常命令専用CPU1aから汎用CPU2bへの実行依頼の通信方式をWadで示している。

【0085】また、通常命令専用CPU1bから通常命令専用CPU1aへの実行依頼の通信方式をWbaで示しており、通常命令専用CPU1bから汎用CPU2aへの実行依頼の通信方式をWbcで、通常命令専用CPU1bから汎用CPU2bへの実行依頼の通信方式をWbdで示している。

【0086】(b)では、各CPU間の6つの実行依頼の通信方式Wab～Wbdを、パラメータとして遺伝子Gを作成しており、“0”はCPU間の実行依頼の通信方式として上記第1実施例のポーリング方式を採用したことを示しており、“1”はCPU間の実行依頼の通信方式として上記第2実施例のインタラプト方式を採用したことを示している。



【0087】そして、シミュレーション装置10は、遺伝子プール（図示せず。）にこのような遺伝子Gを複数記憶しており、この遺伝子Gを後述する遺伝的アルゴリズムによるシミュレーションによって進化させて、最適遺伝子を遺伝子プール上に増大させるように構成されている。

【0088】次に、この第4実施例の動作を図面を参照して説明する。

【0089】図9は、この第4実施例の遺伝的アルゴリズムによるCPU間の実行依頼の通信方式の決定方法を示している。

【0090】シミュレーション装置10では、まず遺伝子プール（図示せず。）中の図8（b）に示す各遺伝子Gを前回の評価に応じて複製し（ステップ400）、続いてこれらの各遺伝子Gを交差させて（ステップ410）、さらにある確率で遺伝子Gに突然変異を起こさせ（ステップ420）、遺伝子Gを進化させる。

【0091】そして、シミュレーション装置10では、遺伝子プール中の全ての遺伝子Gに基づきユーザプログラム実行および周辺処理実行のシミュレーションを行って、プログラム実行時間やサイクルタイムから各遺伝子Gの評価を行う（ステップ430、ステップ440“N”）。  
20

【0092】全ての遺伝子Gについてシミュレーションおよび評価が終了すれば（ステップ440“Yes”）、遺伝子Gが進化するように、このような遺伝子Gの複製、交差、突然変異、全遺伝子Gについてのシミュレーション・評価処理（ステップ400～440）を十分な回数に達するまで繰り返し（ステップ450“N”）、十分な回数に達した場合には（ステップ450  
30 “Yes”）、遺伝子の進化を終了させて、現在最も評価の高い遺伝子Gが、サイクルタイムの最も小さい最適な各CPU間の実行依頼の通信方式を示しているものとして選択する（ステップ460）。

【0093】そして、このようにして選択された遺伝子Gに基づいて、例えば、通常命令専用CPU1aがユーザプログラムから命令を読み出して自CPU用の通常命令を実行している場合に、通常命令専用CPU1b用の通常命令を読み出した場合には、この最適な遺伝子GのWabが示す通信方式によって通常命令専用CPU1bに実行依頼を送出する一方、汎用CPU2b用の特別命令を読み出した場合には、この最適な遺伝子GのWadが示す通信方式によって汎用CPU2bに実行依頼を送出して、各々の命令を実行させるようにする。

【0094】従って、この第4実施例によれば、通常命令専用CPUと汎用CPU間の実行依頼の通信方式をポーリング方式で行うか、あるいはインタラプト方式で行うかを遺伝子で表わし、ユーザプログラム実行前あるいは実行時にこの遺伝子を用いてプログラム実行および周辺処理のシミュレーションを行い、遺伝的アルゴリズム  
50

によってサイクルタイムの最も小さくなる各CPU間の実行依頼の通信方式を試行錯誤的に決定するようにしたため、通常命令専用CPUや汎用CPUが複数台ある場合でも、最適な各CPU間の実行依頼の通信方式を選択できると共に、CPU資源を有効に利用できることになる。

【0095】なお、上記各実施例では、PLCを実施例として本発明のデータ処理装置を説明したが、本発明は、PLCに限定されることはなく、その他一般的なコンピュータのデータ処理装置に適用することができる。

【0096】

【発明の効果】以上説明したように、請求項1～5記載の発明によれば、汎用CPUがプログラム中の特別命令の実行の他に、周辺処理の実行も行うため、CPU資源を有効に利用できる。

【0097】また、請求項3記載の発明によれば、通常命令専用CPU1と汎用CPU2間の実行依頼の通信方式に割り込み信号によるインタラプト方式を採用したため、ユーザプログラムの実行中であっても、汎用CPU2は特別命令を実行してなければ周辺処理を実行することができるため、よりCPU資源を有効に利用できる。

【0098】また、請求4記載の発明によれば、通常命令専用CPUと汎用CPU間の実行依頼の通信方式をポーリング方式で行うか、あるいはインタラプト方式で行うかをユーザプログラム実行時に所定の評価項目にファジイ推論によって推論してサイクルタイムの小さくなる通信方式を決定するようにしたため、ユーザプログラムに応じてユーザニーズにあった、サイクルタイムの小さくなる実行依頼の通信方式を動的に選択できる。

【0099】また、請求項5記載の発明によれば、通常命令専用CPUと汎用CPU間の実行依頼の通信方式をポーリング方式で行うか、あるいはインタラプト方式で行うかを遺伝子で表わし、ユーザプログラム実行前あるいは実行時にこの遺伝子を用いてプログラム実行および周辺処理のシミュレーションを行い、遺伝的アルゴリズムによってサイクルタイムの最も小さくなる各CPU間の実行依頼の通信方式を試行錯誤的に決定するようにしたため、通常命令専用CPUや汎用CPUが複数台ある場合でも、最適な各CPU間の実行依頼の通信方式を選択できる。

【図面の簡単な説明】

【図1】本発明に係るPLCの構成を示すブロック図。

【図2】第1実施例のポーリング方式による汎用CPU2側の処理を示すフローチャート。

【図3】第2実施例のインタラプト方式による汎用CPU2側の実行処理を示すフローチャート。

【図4】第3実施例に係るPLCの構成を示すブロック図。

【図5】（a）～（c）、各々、ファジイ推論の際の各ルール1～3のメンバーシップ関数を示す説明図。

【図6】第3実施例における動作を示すフローチャート。

【図7】第4実施例に係るPLCの構成を示すブロック図。

【図8】(a)、(b)、各々、各CPU間の実行依頼の通信方式に遺伝的アルゴリズムを適用するためのモデルと、このモデルにおける遺伝子を示す説明図。

【図9】第4実施例の遺伝的アルゴリズムによるCPU間の実行依頼の通信方式の決定方法を示すフローチャート。

【図10】従来のデータ処理における並列処理を示す説明図。

【符号の説明】

1 通常命令専用CPU

2 汎用CPU

3 周辺処理専用CPU

4 周辺処理専用CPU

21 特別命令実行部

22 周辺処理部

23 ファジイ推論部

1a 通常命令専用CPU

1b 通常命令専用CPU

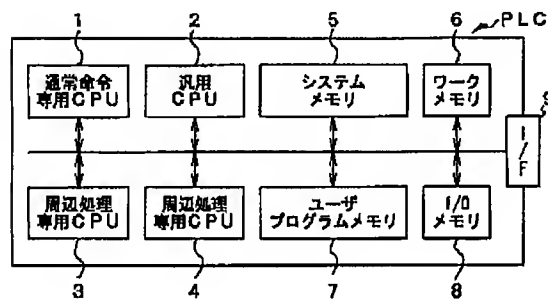
10 2a 汎用CPU

2b 汎用CPU

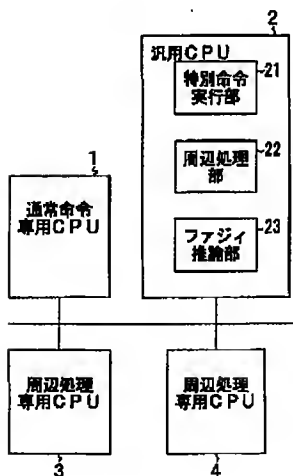
10 シミュレーション装置

G 遺伝子

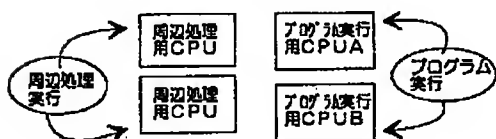
【図1】



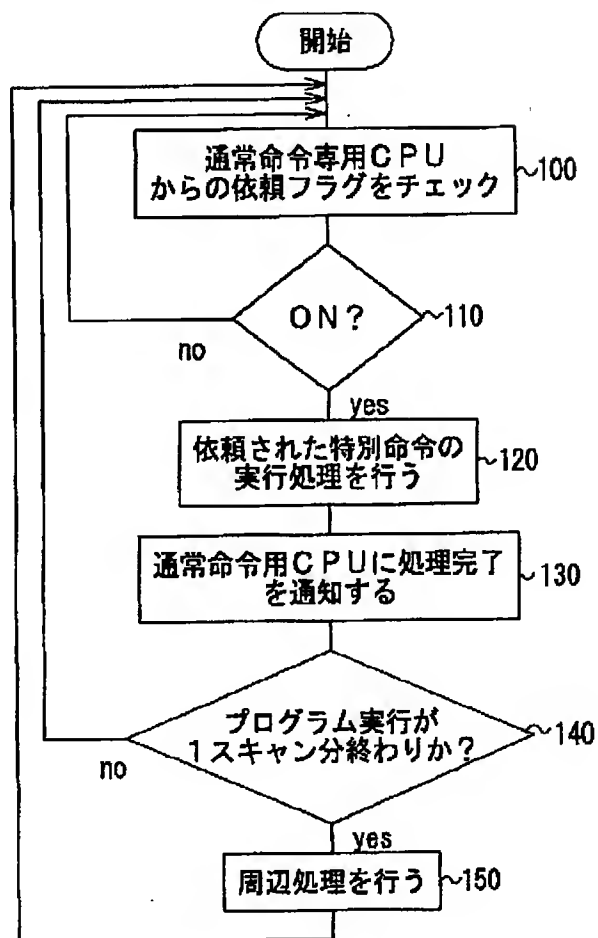
【図4】



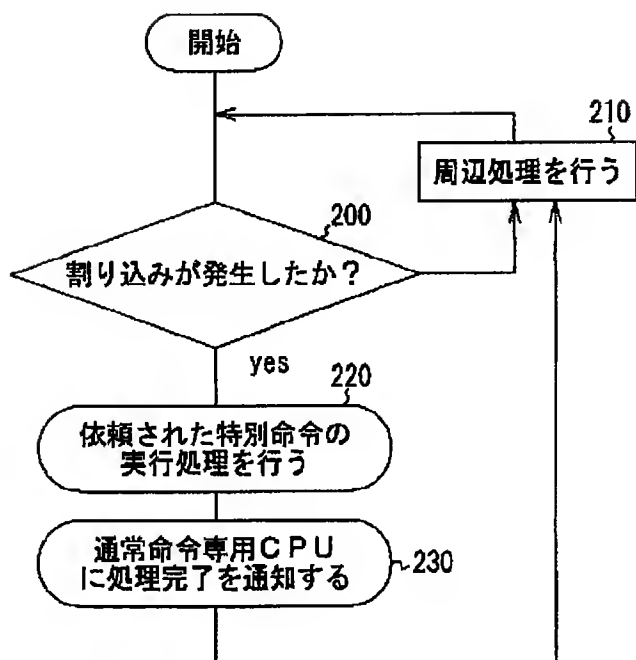
【図10】



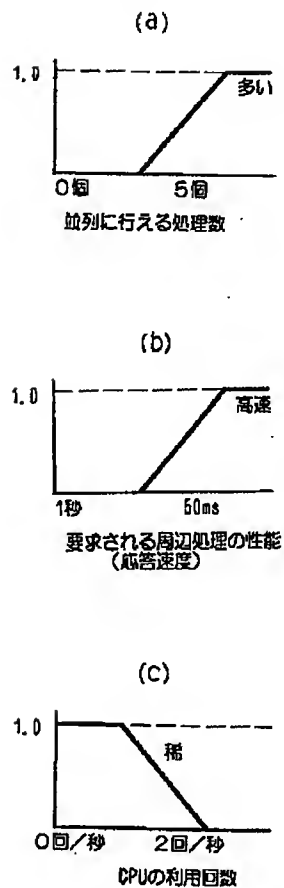
【図2】



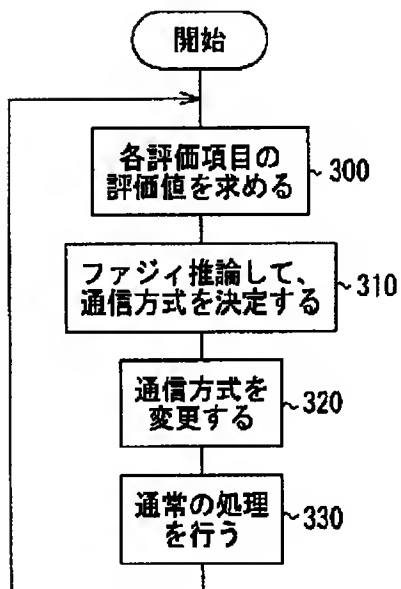
【図 3】



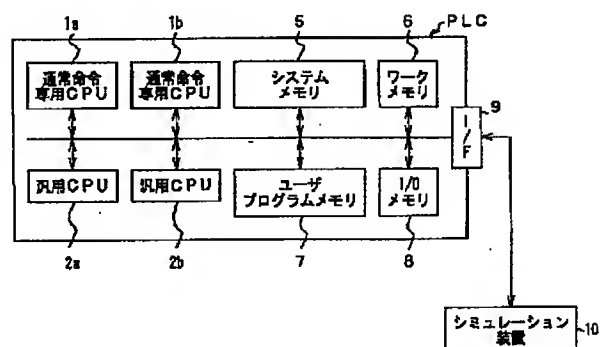
【図 5】



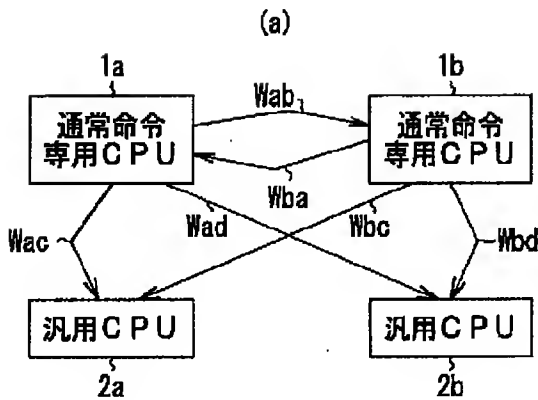
【図 6】



【図 7】



【図 8】



(b)

	G
Wab	1
Wac	0
Wad	1
Wba	1
Wbc	0
Wbd	0

【図 9】

